How To Configure SSL/TLS With Nagios Network Analyzer 2026

Purpose

This document describes how to configure SSL with Nagios Network Analyzer 2026.

Terminology

For your information:

- SSL = Secure Sockets Layer
- TLS = Transport Layer Security

TLS replaces SSL, however the tools used to implement both generally use SSL in their name/directives. For simplicity reasons, the rest of this document will use the term SSL.

To implement SSL you need to generate a certificate. When you generate a certificate, you create a certificate signing request (CSR) that needs to be signed by a Certificate Authority (CA). This CA can be:

- A trusted company like VeriSign, Google, Amazon, COMODO, or Azure
- An internal CA that is part of your IT infrastructure, like Microsoft Windows CA
- The Nagios Network Analyzer server itself (self-signed cert)

The CA will then provide you with a signed certificate.

This documentation can be used to generate a CSR that can be submitted to any of these CA types.

Editing Files

In many steps of this documentation, you will be required to edit files. This documentation will use the vi text editor. When using the vi editor:

- To make changes press `i` on the keyboard first to enter insert mode
- Press **Esc** to exit insert mode
- When you have finished, save the changes in vi by typing `:wq` and press **Enter**

Installing Necessary Components

Establish a terminal session with your Nagios Network Analyzer server and as root and execute the following command:

RHEL | CentOS | Oracle Linux

```
yum install -y mod_ssl openssl
```

Debian | Ubuntu

```
apt install -y openssl
```

Certificate Directory

The steps in this documentation will be performed from within the `/usr/local/nagiosna/var/certs/` directory. Execute the following commands to create the directory (if it doesn't exist) and then change into the directory:

```
mkdir -p /usr/local/nagiosna/var/certs
cd /usr/local/nagiosna/var/certs
```

You will continue to use this terminal session throughout this documentation.

Generate Private Key File

The first step is to generate the private key file, execute the following command:

```
openssl genrsa -out nagiosna.key 2048
```

That will generate some random text.

Generate Certificate Signing Request File

Next you will generate the CSR file by executing the following command:

```
openssl req -new -key nagiosna.key -out nagiosna.csr
```

You will need to supply some values, some can be left blank, however the most important value is the **Common Name**. In the example below you can see that `na-c7x-x64.domain.local` has been used which means that when you access the Nagios Network Analyzer server in your web browser, this is the address you will need to use. This is particularly important, if these don't match then you will get warnings in your web browser. More detailed information about this can be found in the Knowledge Base (KB) article:

SSL/TLS - Understanding Certificate Warnings

The following is an example:

```
Country Name (2 letter code) [XX]:AU
State or Province Name (full name) []: NSW
Locality Name (eg, city) [Default City]: Sydney
Organization Name (eg, company) [Default Company Ltd]: My Company Pty Ltd
Organizational Unit Name (eg, section) []: IT
Common Name (eg, your name or your server's hostname): na-c7x-x64.domain.local
Email Address []:
Please enter the following 'extra' attributes to be sent with your certificate
request
A challenge password []:
An optional company name []:
```

As you can see above, an `email address`, `password` or `optional company name` was not supplied. Specifically, providing a `password` is not necessary.

Sign Certificate Request

At this point you have created a CSR that needs to be signed by a CA.

Using A Trusted CA Company

If you are going to use a trusted company like VeriSign to provide you with a certificate you will need to send them a copy of the certificate request. This can be viewed by executing the following command:

```
cat nagiosna.csr
```

You'll get a lot of random text, this is what you will need to provide to your trusted CA. You must provide the CA with everything including the `-----BEGIN CERTIFICATE REQUEST-----` and `-----END CERTIFICATE REQUEST-----` lines.

Once they send you the signed certificate you will need to copy the certificate into a new file called `nagiosna.crt`. The certificate you receive will also be a lot of random text, so you can just paste that text into the new file which you can open with the vi editor:

```
vi nagiosna.crt
```

You must paste everything including the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` lines when pasting them into the file. Be sure to remove any extra whitespace or lines to avoid errors.

Save the file and close vi.

You can now proceed with the Set Permissions section of this document.

Using A Microsoft Windows CA

If you are going to use a Microsoft Windows CA to sign your CSR, please follow the steps in this KB article:

SSL/TLS - Signing Certificates With A Microsoft Certificate Authority

After following the KB article, you will have the `nagiosna.crt` file and you can proceed to the Set Permissions section of this document.

Self Signing The Certificate

You can also self-sign the certificate by executing the following command:

```
openssl x509 -req -days 365 -in nagiosna.csr -signkey nagiosna.key -out nagiosna.crt
```

Which should produce output saying `Certificate request self-signature ok`.

When you self-sign a certificate you will get warnings in your web browser. More detailed information about this can be found in the following KB article:

SSL/TLS - Understanding Certificate Warnings

Set Permissions

You need to set permissions on the files, and execute the following command:

```
chmod go-rwx nagiosna.*
```

Update Apache Configuration

Certificate

Now you have to tell the Apache web server about the certificate. The configuration file for this differs depending on your operating system (OS), open the SSL file in vi by executing the following command:

RHEL | CentOS | Oracle Linux

```
vi /etc/httpd/conf.d/ssl.conf
```

Debian | Ubuntu

```
vi /etc/apache2/sites-available/default-ssl.conf
```

Find these lines and update them as follows:

```
SSLCertificateFile /usr/local/nagiosna/var/certs/nagiosna.crt
SSLCertificateKeyFile /usr/local/nagiosna/var/certs/nagiosna.key
```

Tip: typing `/SSLCert` and pressing **Enter** in vi should take you directly to this section in the file.

Save the changes; you have finished editing this file.

Enable SSL

You have to update two Apache web server config files to force SSL to be used. The configuration files for this differ depending on your OS, so be sure to use the commands below for your OS.

1. Editing the nna.conf File

Open the `nna.conf` file in vi by executing the following command:

RHEL | CentOS | Oracle Linux

```
vi /etc/httpd/conf.d/nna.conf
```

Debian | Ubuntu

```
vi /etc/apache2/sites-available/nna.conf
```

Entirely replace the text in the `nna.conf` with the following:

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    DocumentRoot /var/www/html/nagiosna/public

    <Directory /var/www/html/nagiosna/public>
        Options +FollowSymLinks
        AllowOverride All
        Require all granted
        DirectoryIndex index.php
    </Directory>

    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
</VirtualHost>
```

Once the file contains only the above lines, save the changes. You have finished editing this file, and can move on to updating the `ssl.conf` file.

Editing the ssl.conf File

Open the `ssl.conf` file in vi by executing the following command:

RHEL | CentOS | Oracle Linux

```
vi /etc/httpd/conf.d/ssl.conf
```

Debian | Ubuntu

```
vi /etc/apache2/sites-available/default-ssl.conf
```

Next, add these lines inside the `VirtualHost` entry:

```
DocumentRoot /var/www/html/nagiosna/public

<Directory /var/www/html/nagiosna/public>
    Options +FollowSymLinks
    AllowOverride All
    Require all granted
    DirectoryIndex index.php
</Directory>

Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"
```

Once you have added those lines, save the changes. You have finished editing this file, and can move on to restarting Apache.

Restart Apache

You need to restart the Apache for the new certificate key to be used.

RHEL | CentOS | Oracle Linux

```
systemctl restart httpd.service
```

Debian | Ubuntu

```
a2ensite default-ssl
a2enmod ssl
a2enmod headers
systemctl restart apache2.service
```

Next, we'll cover firewall rules.

Firewall Rules

The following firewall rules may need to be added. If you cannot access the Nagios Network Analyzer in the next step (Test Certificate) then it's likely you'll need to run these commands:

RHEL | CentOS | Oracle Linux

```
firewall-cmd –zone=public –add-port=443/tcp
firewall-cmd –zone=public –add-port=443/tcp --permanent
```

Debian

The local firewall is not enabled on Debian by default and no steps are required here. **IF** it is enabled then the commands are:

```
iptables -I INPUT -p tcp –destination-port 443 -j ACCEPT
```

Ubuntu

The local firewall is not enabled on Ubuntu by default and no steps are required here. **IF** it is enabled then the commands are:

```
sudo ufw allow https
sudo ufw reload
```

Test Certificate

Now test your connection to the server by directing your web browser to:

```
https://your_server_address/
```

Note: There is no `nagiosna/extension` in the URL, we are just testing a connection to Apache to see if the certificate works.

You may get a self-signed certificate warning, but that is OK, you can just add a security exception. If it is working, you'll see the Nagios Network Analyzer welcome page. More detailed information about this can be found in the following KB article: SSL/TLS - Understanding Certificate Warnings.

If it returns an error check your firewall and backtrack through this document, making sure you've performed all the steps listed.

You are now set to use `https` with your Nagios Network Analyzer web interface.

Notes On Redirecting

With this configuration, if a user types `http://nagiosna` in their web browser, it will redirect them to `https://nagiosna` which can cause certificate warnings in certain scenarios. If you wanted to redirect them to `https://nagiona.yourdomain.com` then you simply need to change the `RewriteRule` in the `/etc/httpd/conf/httpd.conf` file:

```
RewriteRule (.*) https://nagiosna.yourdomain.com%{REQUEST_URI}
```

Then restart the `httpd` service.

More detailed information about this can be found in the following KB article:

SSL/TLS - Understanding Certificate Warnings

Finishing Up

This completes the documentation on how to configure SSL on your Nagios Network Analyzer. If you have additional questions or other support-related questions, please visit us at our Nagios Support Forum, Nagios Knowledge Base, or Nagios Library:

Visit Nagios Support Forum Visit Nagios Knowledge Base Visit Nagios Library